Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Matej Biberović

Pametna ključavnica NFC

diplomsko delo

visokošolski strokovni študijski program prve stopnje računalništvo in informatika

Mentor: doc. dr. Mira Trebar

Ljubljana 2015

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:
Pametna ključavnica NFC

Tematika naloge:

Tehnologija NFC (Near Field Communication) je vse bolj razširjena in omogoča razvoj številnih rešitev v povezavi z mobilnimi napravami. Kandidat naj v diplomskem delu zasnuje ključavnico, ki bo delovala na osnovi komunikacije kratkega dosega. Pametni telefon NFC nadomesti ključ za odklepanje. Razvije in implementira naj mobilno aplikacijo za operacijski sistem Android, ki zagotavlja varno delovanje elektronskega sistema z uporabo značke NFC. Rešitev naj zagotavlja funkcionalnosti za odklepanje, registracijo uporabnikov, beleženje dostopov in hranjenje zgodovine.

Izjava o avtorstvu diplomskega dela

Spodaj podpisani Matej Biberović sem avtor diplomskega dela z naslovom:

Pametna ključavnica NFC

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Mire Trebar,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, dne 28. septembra 2015 Podpis avtorja:

Iskreno se zahvaljujem svoji punci, staršem, prijateljem, in vsem, ki so mi nudili kakršnokoli pomoč in podporo v času študija.
Posebna zahvala gre doc. dr. Miri Trebar, za nasvete ter strokovno pomoč pri pisanju diplomskega dela.

Svoji dragi Urški

Kazalo

Seznam uporabljenih kratic

kratica	opis
AES	Advanced Encryption Standard
API	Application Programming Interface
BT	Bluetooth
GPS	Global Positioning System
HTTP	Hyper Text Transfer Protocol
ISO	International Organization for Standardization
JSON	JavaScript Object Notation
NDEF	NFC Data Exchange Format
NFC	Near Field Communication
REST	REpresentational State Transefer
RFID	Radio Frequency IDentification
SHA	Secure Hash Algorithm
SOAP	Simple Object Access Protocol
SQL	Structured Query Language
TI	Texas Instruments

Povzetek

Z uporabo tehnologije za komunikacijo kratkega dosega - NFC (Near Field Communication) je bil izdelan sistem, ki združuje elektronsko ključavnico z vgrajenim mikrokrmilnikom, mobilno aplikacijo ter spletno storitev kot podporo delovanju aplikacije. Sistem nam omogoča odklepanje elektronske ključavnice s pametnim telefonom, posojanje dostopa drugim uporabnikom (za določen čas) ter beleženje dostopa s časovno značko in lokacijo. Za preprečevanje nelegitimnih dostopov ima sistem implementirane varnostne mehanizme s pomočjo simetrične in asimetrične kriptografije ter zgoščevanja. V diplomskem delu je uporabljen mikrokrmilnik TI serije MSP430 in polpasivna NFC značka TI RF430. Spletna storitev se izvaja na aplikacijskem strežniku GlassFish, mobilna aplikacija pa na operacijskem sistemu Android. Za zagotavljanje varnosti so implementirani kriptografski algoritmi AES, RSA in SHA.

Ključne besede: NFC, elektronska ključavnica, kriptografija, Android, mikrokrmilnik, spletna storitev.

Abstract

With the use of short range contactless technology - NFC (Near Field Communication) a system, which contains an electronic lock with microcontroller, a mobile application and a web service, was developed. The system enables us to unlock an electronic lock with a smartphone, to give temporary access to other people and to log the date and the location of lock access. To prevent illegitimate access, the system implements security mechanisms with the help of symmetric and asymmetric cryptography and hashing. In the thesis, TI series MSP430 microcontroller and TI RF430 semi-passive NFC tag are used. Web service runs on GlassFish application server and mobile application runs on Android operating system. AES, RSA and SHA cryptographic algorithms are implemented to ensure security.

Keywords: NFC, electronic lock, cryptography, Android, microcontroller, web service.

Poglavje 1

Uvod

V sodobnem času tehnološkega napredka, si je skoraj nemogoče predstavljati vsakdanje življenje brez mobilnega telefona. Ker jih imamo skoraj vedno pri sebi, so le-ti uporabljeni za vse več dnevnih opravil. Naše vsakodnevno opravilo je tudi odklepanje ene ali več ključavnic z različnimi ključi, kar bi lahko v bližnji prihodnosti nadomestili z mobilnim telefonom.

S tem bi rešili več problemov v zvezi s ključi: ne bi nam bilo potrebno plačevati za izdelavo dodatnih ključev; za posojanje ključa ne bi bilo potrebne fizične prisotnosti; v primeru izgube ključa, ne bi bilo potrebno menjati ključavnice. Šop ključev bi zamenjali z enim, univerzalnim ključem na mobilnem telefonu.

Najprimernejša tehnologija za izdelavo takšne ključavnice je komunikacija kratkega dosega (NFC - Near Field Communication), ki se uporablja tudi za brezgotovinsko plačevanje. Največji delež mobilnih telefonov s tehnologijo NFC predstavljajo pametni telefoni z operacijskim sistemom Android. Operacijski sistem iOS pa uporablja tehnologijo NFC samo za brezgotovinsko plačevanje ApplePay.

Glede na podane možnosti, je bila zamišljena celovita rešitev, ki zajema elektronsko ključavnico, mobilno aplikacijo in spletno storitev. Elektronska ključavnica je pol-pasivna NFC značka z mikrokrmilnikom, ki se odziva na ukaze, ki jih prejme od mobilnega telefona preko NFC. Mobilna aplikacija

omogoča varno komunikacijo s ključavnico, urejanje ključavnic, v povezavi s spletni storitvijo pa tudi pregledovanje dostopov in dodeljevanje začasnih dostopov.

V nalogi je opisana zasnova elektronske ključavnice od načrta do končnega izdelka, izdelava mobilne aplikacije za operacijski sistem Android v programskem jeziku Java, načrtovanje varnega komunikacijskega protokola in izdelava spletne storitve v Java EE.

Poglavje 2

Uporabljene tehnologije in orodja

2.1 Tehnologije

2.1.1 NFC

Near Field Communication (NFC) je specifikacija za komunikacijo kratkega dosega med dvema napravama. Temelji na tehnologiji uporabljeni za RFID in je standardizirana v ISO/IEC 18092. Omejena je na razdalje do 10cm in namenjena poenostavitvi opravljanja transakcij, izmenjave digitalne vsebine in povezovanju elektronskih naprav na dotik [1]. NFC deluje na frekvenci 13.56 MHz in je bil razvit v sodelovanju med NXP Semiconductors (prej Phillips) in Sony Corporation. Ker ima NFC možnost branja in pisanja na naprave se domneva, da bo imel v prihodnosti širši namen uporabe kot standardne pametne kartice.

NFC sestavljata pobudnik (ang. initiator) in tarča (ang. target). Pobudnik generira radio frekvenčni (RF - Radio Frequency) signal in kontrolira prenos podatkov, tarča pa se odziva na zahteve pobudnika. NFC protokol loči tudi med dvema načinoma komunikacije: aktivni in pasivni. Pri aktivni komunikaciji, tako pobudnik kot tarča komunicirata z generiranjem lastnega

električnega polja. Komunikacija poteka v pol-dupleks načinu: ena stran izključi svoj RF signal dokler druga naprava ne preneha s pošiljanjem podatkov. V tem načinu imata po navadi obe napravi svoj vir energije. Pasivni način komunikacije je pogostejši, kjer pobudnik generira RF signal, tarča pa modulira obstoječe RF polje. Pobudnik te spremembre posluša in procesira podatke. Trenutno podprte hitrosti prenosa podatkov so 106, 212, 424 in 848 kb/s.

Možnost uporabe tehnologije NFC so skoraj neomejene. Med najpogostejšimi primeri uporabe so: identifikacija uporabnika, brezgotovinsko plačevanje, vstopnice, prijava delovnega časa, javni prevoz, fizični dostop, povezovanje naprav in prenos manjših količin podatkov.

Drugi mednarodni standardi, ki ravno tako opisujejo NFC komunikacijo so ISO/IEC 14443, ISO/IEC 15693 in ISO 18000-3.

NFC značke

NFC značka (ang. NFC tag) je pasivna NFC naprava za shranjevanje manjše količine podatkov. Sestavljena je z enostavnega vezja, ki vsebuje mikrokrmilnik, pomnilnik in anteno. Z aktivno NFC napravo, kot je mobilni telefon, lahko z nje beremo zapisane podatke, če kartica ni označena samo za branje, pa lahko vanjo podatke tudi zapisujemo. Značke so najpogosteje v obliki kartic, nalepk ali obeskov za ključe.

Združenje NFC Forum definira štiri različne vrste značk, ki se razlikujejo po velikosti pomnilnika, hitrosti komunikacije, varnosti, obstojnosti podatkov in številu možnih prepisov [2].

Varnost

Čeprav je doseg komunikacije pri tehnologiji NFC zelo omejen, specifikacije ne zagotavljajo varne komunikacije med napravama. Za varnost moramo poskrbeti sami, na višjem, aplikacijskem nivoju z uporabo kriptografskih algoritmov [4]. Znanih je več možnih napadov na tehnologijo NFC:

Prisluškovanje Ker uporablja NFC brezžično komunikacijo je prisluškovanje velik problem. Ko dve napravi komunicirata prek NFC, za komunikacijo uporabljata RF signale. Napadalec lahko uporabi anteno za prestrezanje tega signala, in z ustreznim znanjem izlušči poslane podatke. Največja razdalja, s katere je možno prisluškovanje je odvisna od večih dejavnikov in obsega od 1m za pasivno do 10m za aktivno komunikacijo. S šifriranjem poslanih podatkov se ne zavarujemo pred prisluškovanjem, vendar napadalec pridobljenih podatkov ne razume.

Ponavljanje Pri napadu s ponavljanjem napadalec tako kot pri prisluškovanju zajame promet in ga kasneje spet pošlje. Pri zajemu pravih podatkov lahko napadalec pridobi nepooblaščen dostop. Da preprečimo napad s ponavljanjem, ob vsaki komunikaciji prenesemo enkratno generirano naključno število (ang. nounce), ki mora ostati isto v eni seji. Druga stran s tem številom kriptografsko zgosti vsebimo (npr. ključ) in to pošlje nazaj. Če se naključno število med komunikacijo spremeni, sejo prekinemo [3].

Okvara podatkov Namesto samega prisluškovanja komunikaciji, lahko napadalec poskuša spremeniti prenešene podatke. V najenostavnejšem primeru napadalec samo moti komunikacijo, tako da sprejemnik ne razume poslanih podatkov. Okvaro podatkov lahko dosežemo s pošiljanjem pravih frekvenc ob pravem času, ki ga lahko izračunamo, če poznamo podrobnosti komunikacijskega protokola. Napad ni zapleten, vendar ne dovoljuje napadalcu spreminjanja samih podakov.

Spreminjanje podatkov Pri spreminjanju podatkov napadalec želi, da naprava ki sprejema, prejme veljavne, a spremenjene podatke. Izvedljivost napada je zelo odvisna od uporabljene moči amplitudne modulacije. Pri 100% modulaciji je možno spremeniti le nekatere bite, pri 10% modulaciji pa je možno spremeniti vse bite. Spreminjanje podatkov lahko preprečimo z uporabo varnega prenosa podatkov [4].

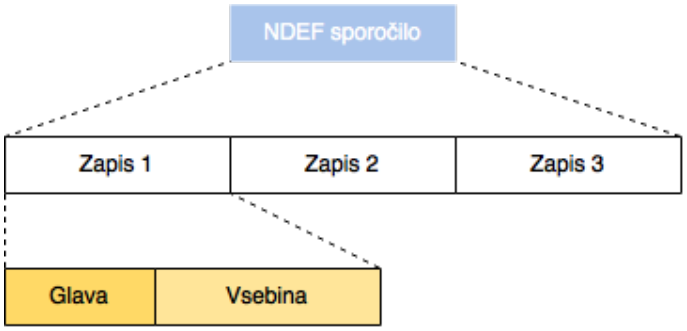
Vrivanje podatkov Pri tem napadu napadalec vrine svoja sporočila v komunikacijo med dvema napravama. Ta napad je možen le v primeru, da druga naprava potrebuje zelo veliko časa za odgovor. Napadalec bi lahko v tem času vrinil svojo sporočilo. Vrivanje bi bilo uspešno le v primeru, da so podatki poslani prej kot bi druga naprava odgovorila. V nasprotnem primeru bi se signali pomešali in prišlo bi do okvare podatkov.

Vrinjeni napadalec (ang. MITM - Man-in-the-Middle) Pri MITM napadu se napadalec vrine v komunikacijo med dvema napravama, ki se ne zavedata, da ne komunicirata med seboj. Napadalec tako prestreza vso komunikacijo in jo posreduje naprej, vsebino pa lahko poljubno spreminja. Ta napad v scenariju s tehnologijo NFC, zaradi težavnosti izvedbe ni možno izvesti [4].

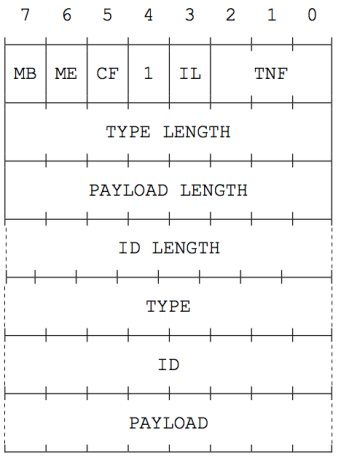
NDEF

Sporočila, ki se prenašajo s pomočjo tehnologije NFC so lahko zapisana v različnih formatih. Najbolj razširjen je format NDEF (NFC Data Exchange Format), ki ga je razvilo združenje NFC Forum [5]. NDEF je specifikacija, ki definira format sporočil, za prenos med dvema NFC napravama. To je enostaven binarni format, ki ga lahko uporabimo za enkapsulacijo enega ali več podatkov različnega tipa in dolžine v eno samo sporočilo. NDEF sporočilo je sestavljeno iz enega ali več NDEF zapisov (slika 2.1). Največje število zapisov je odvisno od aplikacije in vrste značke.

NDEF zapis ima ločeno glavo in vsebino, kjer se nahaja dejansko sporočilo (slika 2.2). Obstaja kratek in dolg NDEF zapis. Kratek zapis se uporablja za zapise z vsebino do 255 bajtov, dolg pa za zapise z vsebino do 2^{32}-1 bajtov [6]. Tabela 2.1 opisuje posamezna polja v zapisu.



Slika 2.1: Zgradba NDEF sporočila



Slika 2.2: Zgradba kratkega NDEF zapisa [6]

Polje	**Dolžina**	**Opis**
MB (Message Begin)	1 bit	Prvi zapis v NDEF sporočilu
ME (Message End)	1 bit	Zadnji zapis v NDEF sporočilu
CF (Chunk Flag)	1 bit	Prvi ali vmesni del bloka
SR (Short Record)	1 bit	Vsebina krajša od 256 bajtov
IL (ID_LENGTH present)	1 bit	Vsebuje ID_LENGTH in ID
TNF (Type Name Format)	3 biti	Vrsta vsebine
TYPE_LENGTH	8 bitov	Velikost polja TYPE
PAYLOAD_LENGTH	8/32 bitov	Velikost vsebine
ID_LENGTH	8 bitov	Velikost polja ID
TYPE	n bitov	Vrsta vsebine (opisno)
ID	n bitov	Identifikator vsebine
PAYLOAD	n bitov	Vsebina zapisa

Tabela 2.1: Opis polj NDEF zapisa

2.1.2 Spletne storitve

Spletne storitve predstavljajo standardiziran način prenosa podatkov prek obstoječih internetnih protokolov, kar nam omogoča povezljivost z različnimi vrstami aplikacij. Programi, napisani v različnih programskih jezikih, lahko tako uporabljajo iste spletne storitve. Vrnjeni podatki so največkrat v formatu JSON ali XML. Primer spletne storitve je pridobivanje tečajne liste Banke Slovenije. Različne aplikacije lahko tako dobijo dnevno sveže podatke o menjalnih tečajih.

REST

REST je arhitekturni stil za načrtovanje spletnih storitev. Ideja je, da namesto zapletenih mehanizmov za povezavo naprav, kot je na primer SOAP (Simple Object Access Protocol), uporabimo enostaven HTTP (Hyper Text Transfer Protocol) protokol. Spletni storitvi, ki se drži teh dogovorov, pravimo, da je RESTful spletna storitev. RESTful storitve uporabljajo HTTP

zahteve za pošiljanje (PUT, POST), branje (GET) in brisanje (DELETE) podatkov. Podatki so najpogosteje vrnjeni v formatu JSON [7].

Java EE, GlassFish

Java EE (ang. Enterprise Edition) je verzija Jave, namenjena razvoju spletnih aplikacij in storitev. Je nadgradnja Java SE (ang. Standard Edition) in zagotavlja API (Application Programming Interface) za objektno-relacijske preslikave, razdeljene in večslojne arhitekture ter spletne storitve.

Privzeti strežnik za Java EE spletne storitve je GlassFish. To je razširljiv, odprtokodni aplikacijski strežnik in je testno povezan z Java EE. Na voljo je tudi dodatek Jersey, ki poenostavi razvoj RESTful spletnih storitev.

Java Beans objekti

Java Beans objekti so enostavni Javanski razredi, ki se držijo sporazuma o poimenovanju in obnašanju. Razred mora imeti privzeti konstruktor brez argumentov, lastnosti morajo biti dostopne z metodami `set`, `get` in `is` ter mora omogočati serializacijo. Takšni objekti omogočajo enostavno pretvorbo v JSON obliko zapisa in nazaj.

Statusne kode odgovorov

Odgovor na vsako zahtevo, ki jo posredujemo spletni storitvi vsebuje statusno kodo HTTP (ang. HTTP status code). To je trimestna koda, ki določa stanje poslane zahteve. Delijo se na 5 vrst, kjer ima vsaka vrsta svojo predpono:

- `1xx`: Informativne
- `2xx`: Zahteva uspešna
- `3xx`: Preusmeritev
- `4xx`: Napaka pri zahtevi
- `5xx`: Napaka na strežniku

Izmed vseh statusnih kod se pri RESTful storitvah največ uporabljajo naslednje:

- `200 OK`: Zahteva je uspešna.
- `201 Created`: Zahteva je uspešna in ustvarjen je nov vnos.
- `204 No content`: Zahteva je uspešna in ni vrnjene vsebine.
- `400 Bad Request`: Zahteve ni bilo možno prepoznati zaradi napačne sintakse.
- `401 Unauthorized`: Zahteva nima pravice za izvršitev. Manjka generiran žeton v glavi zahteve.
- `403 Forbidden`: Zahteva ima prepovedan dostop.
- `404 Not Found`: Pot za poslano zahtevo ne obstaja.

2.1.3 Podatkovna baza

SQLite

SQLite je odprtokodna vgrajena (ang. embedded) relacijska podatkovna baza, izdana leta 2000. Zasnovana je tako, da aplikacijam zagotavlja enostaven in priročen način za upravljanje podatkov, za razliko od večjih namenskih sistemov za upravljanje s podatkovnimi bazami. SQLite podatkovna baza ima ugled, da je prenosna, enostavna za uporabo, kompaktna, učinkovita in zanesljiva [8].

Podpora za SQLite je že vgrajena v operacijski sistem Android in nudi ustrezne razrede za njegovo uporabo.

Java DB

Java DB je odprtokodna distribucija podatkovne baze Apache Derby, podprta s strani podjetja Oracle. Podpira standarde ANSI/ISO SQL preko

JDBC (Java Database Connectivity) in Java EE APIjev. Zaradi kompaktne velikosti (2.6 MB) je vključena v razvijalski paket in jo lahko vgradimo v jedro aplikacije. Java DB je relacijska podatkovna baza.

2.2 Strojna oprema

2.2.1 Mikrokrmilnik MSP430G2231

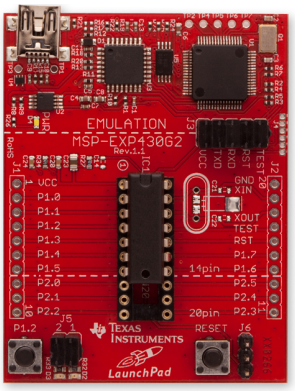
MSP430G2231 je nizkocenovni mikrokrmilnik iz družine MSP430 podjetja Texas Instruments. MSP430 je družina mikrokrmilnikov z zelo nizko porabo in posledično dolgo avtonomijo baterije, zato je primerna za uporabo v prenosnih vgrajenih sistemih. Naprava ima zmogljiv 16 bitni procesor in 16 bitne naslovne registre. Digitalno krmiljen oscilator pa omogoča prebujanje iz načina nizke porabe v manj kot 1 mikrosekundi. Cena na kos pri nakupu tisoč kosov je približno 0.55 dolarjev [9].

Programator MSP-EXP430G2 LaunchPad

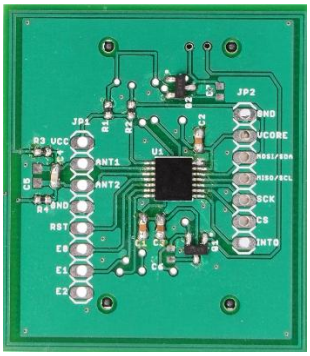
MSP-EXP430G2 je enostaven programator in razhroščevalnik za mikrokrmilnike iz družine MSP430. Priključimo ga na računalnik z USB kablom, mikrokrmilnik pa programiramo s klikom na gumb Flash v razvojnem okolju Code Composer Studio. Prikazan je na sliki 2.3.

2.2.2 Dinamična NFC značka RF430CL330H

RF430CL430CL330H podjetja Texas Instruments je dinamična NFC značka tipa 4. Komunikacija z mikrokrmilnikom poteka prek serijskega protokola SPI ali I^2C. Ima 3KB SRAM pomnilnika za NDEF sporočila. Omogoča avtomatsko preverjanje pravilnosti zapisanega NDEF sporočila in proženje prekinitev ob koncu branja, koncu pisanja in napakah. Cena na kos pri nakupu tisoč kosov je približno 0.85 dolarjev [10]. Na sliki 2.4 je prikazana značka na vezju, ki že vsebuje potrebne elemente za delovanje (integrirano vezje, antena, upori, kondenzatorji).



Slika 2.3: Programator za mikrokrmilnike serije MSP430



Slika 2.4: Dinamična NFC značka RF430CL330H z anteno

2.3 Programska oprema

2.3.1 Android

Android je programska platforma in operacijski sistem (OS) za mobilne naprave, ki temelji na operacijskem sistemu Linux in je razvit s strani podjetja Google in Open Handset Alliance. Razvijalcem omogoča pisanje aplikacij v programem jeziku Java in uporabo knjižic razvitih za Android.

Za objavo in prenos razvitih aplikacij se uporablja spletna trgovina Google Play, na katero je naloženih že milijon šesto tisoč aplikacij [12].

Android je vodilni OS med pametnimi telefoni in obsega 78.0% tržni delež. Sledi mu operacijski sistem iOS podjetja Apple z 18.3% tržnim deležem (prvo čertletje 2015) [11].

Programske knjižice

Za Android je dostopnih veliko brezplačnih in odprtokodnih programskih knjižnic, ki nam olajšajo razvoj mobilnih aplikacij. Večina izvorne kode je dostopna na portalu `github.com`. V nalogi so uporabljene naslednje programske knjižice:

Retrofit omogoča enostavno komunikacijo z RESTful spletno storitvijo s pomočjo Javanskih vmesnikov.

Picasso omogoča enostavno asinhrono nalaganje slik iz spletnih strežnikov v uporabniški vmesnik.

Material Calendar View omogoča prikaz izbire datuma v uporabniškem vmesniku.

BouncyCastle omogoča integracijo varnostnih fukcij (zgoščevanje, šifriranje, podpisovanje).

2.3.2 Integrirana razvojna okolja

Android Studio je integrirano razvojno okolje za platformo Andrioid, ki temelji na razvojnem okolju IntelliJ IDEA. Googlov produktni vodja, Ellie Powers, ga je 16. maja 2013 objavil na Googlovi I/O konferenci. Zasnovan je posebej za razvoj Android aplikacij in omogoča takojšnji predogled izgleda aplikacije ter napredno dokončevanje kode. Deluje na operacijskih sistemih Windows, Mac OS X in Linux.

Code Composer Studio (CCS) je integrirano razvojno okolje, ki podpira krmilnike podjetja Texas Instruments. CCS sestavlja skupek orodij za razvoj in razhroščevanje vgrajenih sistemov. Vsebuje C/C++ prevajalnik, urejevalnik izvorne kode, razhroščevalnik, orodje za spremljanje delovanja (ang. profiler) in še veliko funkcij.

NetBeans je integrirano razvojno okolje, namenjeno predvsem razvoju v programskem jeziku Java, podpira pa tudi PHP, C/C++ in HTML5. Omogoča povezavo s spletnimi storitvami, strežniki in podatkovnimi bazami. Napisan je v programskem jeziku Java in deluje na operacijskih sistemih Windows, OS X, Linux in vseh ostalih, ki podpirajo Javanski navidezni stroj.

2.4 Kriptografija

Kriptografija nam omogoča skrivanje informacij. Uporablja se za šifriranje telefonskih pogovorov, elektronskih sporočil, bančnih transakcij, PIN kod, gesel in spletnih transakcij. Poleg tega se uporablja še za različne informacijske varnostne probleme, vključno z elektronskimi podpisi, ki se uporabljajo za dokazovanje izvirnosti sporočila [13].

2.4.1 Simetrična kriptografija

Pri simetrični kriptografiji se tako za šifriranje, kot tudi za dešifriranje sporočil uporablja en ključ. Algoritmi za simetrično kriptografijo so hitrejši in enostavnejši od algoritmov za asimetrično kriptografijo, vendar je pomankljivost tega sistema, da si morata pred šifriranjem strani izmenjati ključ na varen način. Z algoritmom za izmenjavo ključa Diffie-Hellman si lahko dve strani izmenjata ključ za šifriranje in dešifriranje. Najpogosteje uporabljene implementacije simetrične kriptografije so AES (Advanced Encryption Standard), DES (Data Encryption Standard) in 3DES (Triple Data Encryption Standard).

2.4.2 Asimetrična kriptografija

Asimetrična kriptografija je nadgradnja simetrične kriptografije, ki uporablja dva ključa: zasebni in javni. Oba ključa se generirata hkrati, pri čemer lahko javni ključ izve vsak, zasebni ključ pa moramo varovati pred odtujitvijo. Če želi oseba A šifrirati sporočilo, da bo vidno samo osebi B, mora uporabiti javni ključ osebe B za šifriranje. To sporočilo lahko nato dešifrira samo oseba B s svojim zasebnim ključem. Najpogosteje uporabljen algoritem za asimetricno kriptografijo je RSA.

2.4.3 Zgoščevalne funkcije

Zgoščevalna funkcija je algoritem, ki preslika poljubno dolg niz znakov na vhodu v niz fiksne dolžine na izhodu (ga zgosti), pri čemer zelo majhna sprememba na vhodu povzroči velike spremembe na izhodu. Uporabljajo se na primer pri zgoščevalnih tabelah, primerjanju enakosti dveh datotek, pravilnosti prenosa podatkov in shranjevanju uporabniških gesel. Da je zgoščevalna funkcija kriptografska, ne sme biti obrnljiva oziroma mora biti pridobivanje vhodnega podatka iz izhoda zelo oteženo.

Ker je množica vhodnih podatkov neskončna, možnih izhodov funkcije pa je 2^n (n je dolžina izhoda v bitih), nam Dirichletov princip zagotavlja,

da bo prišlo do trkov (ang. collisions). To pomeni, da se pri dveh ali več različnih vhodih funkcije izračuna isti izhod. Lastnost dobrih zgoščevalnih funkcij je odpornost proti trkom, kar pomeni, da je zelo težko najti dva različna vhodna podatka z enakim izhodom funkcije. MD5 (Message Digest) je primer kriptografske zgoščevalne funkcije, ki zaradi pomanjkljivosti v kodi ni odporna na napad s trki [15].

Poglavje 3

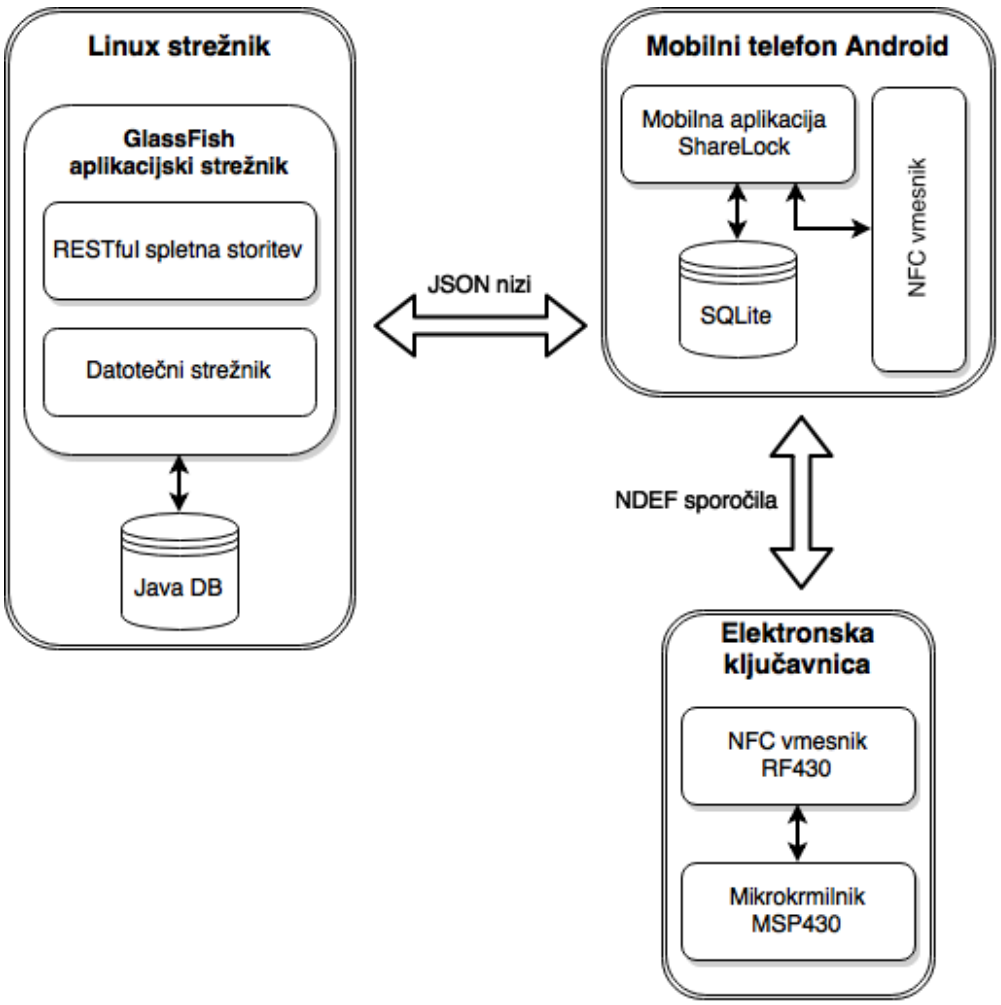
Načrtovanje

Celotna rešitev je sestavljena iz treh delov: elektronske ključavnice ter mobilne aplikacije, ki je podprta s spletno storitvijo. Poenostavljena arhiterktura sistema je predstavljena na sliki 3.1.

Elektronska ključavnica je sestavljena iz ohišja, mehanizma za odklep ter elektronike, ki vključuje baterijo, mikrokrmilnik in dinamično NFC značko. Ko telefon približamo ključavnici, ta po varni povezavi z NFC pošilja ukaze do ključavnice, ki se nanje odziva in se ob pravilnem ukazu tudi odklene. Format sporočil, ki se prenaša med telefonom in ključavnico je NDEF.

Mobilna aplikacija omogoča uporabo večjega števila ključavnic, saj ima vsaka svojo identifikacijsko številko. Omogočeno je tudi dodeljevanje pravice odklepanje drugim uporabnikom za vnaprej določeno časovno obdobje. Možno je pregledovanje zgodovine dostopov s časovno značko in točno GPS lokacijo. Mobilna aplikacija za zagotavljanje dodatnih funkcionalnosti kounicira preko interneta s spletno storitvijo.

Cilj rešitve je čim višja varnost in tudi čim nižja cena izdelka ter minimalna odvisnost od storitev, ki niso vsepovsod na voljo, kot je na primer internet. Internetna povezava zato ni pogoj za uporabo ključavnice, omogoča pa uporabo dodatnih funkcij, ki so opisane v naslednjih poglavjih.



Slika 3.1: Arhitektura sistema

3.1 Obstoječe rešitve

Obstaja že nekaj podobnih pametnih elektronskih ključavnic, vendar je podpora tehnolofije NFC zelo redka. Večina podobnih rešitev ima podporo le za Bluetooth in WiFi povezavo. Primerjava obstoječih rešitev je opisana v tabeli 3.1.

Ime	NFC	BT/WiFi	Deljenje	Zgodovina	Cena
August		✓	✓	✓	$249
Danalock		✓		✓	$159
Genie		✓	✓	✓	$249
Goji		✓	✓		$278
Haven		✓			$219
Kwikset Kevo		✓	✓	✓	$219
Lockitron Bolt		✓	✓		$99
Sesame		✓	✓	✓	$149
Yale Real Living	✓		✓	✓	$225
Key2Share	✓		✓	✓	?
ShareLock	✓		✓	✓	

Tabela 3.1: Primerjava obstoječih rešitev [14]

3.2 Spletna storitev

Ključavnica za enostavno delovanje ne potrebuje internetne povezave, potrebna pa je za povezovanje s spletno storitvijo, ki nam omogoča uporabo dodatnih funkcij. Spletna storitev je povezana s podatkovno bazo, ki hrani podatke o uporabnikih in ključavnicah (dostopi, zgodovina). Spletna storitev nam omogoča:

- Upravljanje z uporabniki (registracija, prijava, urejanje, brisanje, prijatelji)

- Upravljanje s ključavnicami (dodajanje, brisanje, urejanje)
- Upravljanje z začasnimi dostopi (dodajanje, brisanje, ogled)
- Upravljanje z zgodovino dostopov (dodajanje, ogled)

Podrobnejši opis storitev je v poglavju 4.2.

3.3 Varnost

Stopnja varnosti vsake ključavnice je skoraj v celoti odvisna od ključa. Zato je pri naši rešitvi zelo pomembna varna komunikacija med ključavnico in mobilnim telefonom, saj se med njima prenaša ključ, ki je potreben za odklep.

V primeru, da napadalec prisluškuje komunikaciji lahko pridobi glavni ključ, ki omogoča odklep ključavnice. Da se temu izognemo, nad sporočilom izvedemo kriptografsko zgoščevalno funkcijo. Vendar zgoščevanje samo ni dovolj, saj je napad s ponavljanjem še vedno možen. Da se izognemo napadu s ponavljanjem, ključavnica po vsakem dostopu generira naključno število (ang. nounce) in ga shrani na NFC značko. Ta naključno število skupaj z glavnim ključem zgostimo in ga pošljemo nazaj. Če se zgoščeno naključno število in glavni ključ ujemata s poslanim sporočilom, se ključavnica odklene. Podrobnejši opis komunikacije je v poglavju 3.5.

Ker so ključi in uporabniška gesla shranjena v podatkovni bazi na spletnem strežniku, obstaja tudi možnost odtujitve celotne baze. Da onemogočimo napadalcu, da bi izkoristil te podatke so le-ti zaščiteni. Uporabniško geslo je skupaj s soljo (ang. salt) zgoščeno s kriptografsko zgoščevalno funkcijo `SHA-256`. Tako se zavarujemo pred napadi z mavričnimi tabelami in surovo silo (ang. bruteforce attack). Ključ za odklep ključavnice je šifriran s simetričnim šifrirnim algoritmom AES (Advandced Encryption Standard), tako, da ga lahko dešifrira le uporabnik s svojim geslom. Večji problem je deljenje ključa za začasni dostop, ker našega šifriranega ključa drug uporabnik ne bo mogel dešifrirati. Tu je potrebno uporabiti asimetrični šifrirni algoritem RSA, ki uporablja zasebni in javni ključ. Ključ pred pošiljanjem tako

šifriramo z javnim ključem drugega uporabnika. Uporabnik, ki mu dodelimo dostop lahko nato dešifrira ključ s svojim zasebnim ključem.

3.4 Prijava ključavnice

Ključavnici je priložena tudi pasivna NFC značka, na kateri sta zapisani dve vrednosti, ki sta potrebni za odklep in upravljanje ključavnice. Vsaka vrednost je svoj NDEF zapis v NDEF sporočilu. Enake vrednosti so zapisane tudi v pomnilniku ključavnice. Vrednosti:

- **id:** Enolična identifikacijska številka ključavnice
- **key:** Glavni ključ, uporabljen za odklep ključavnice

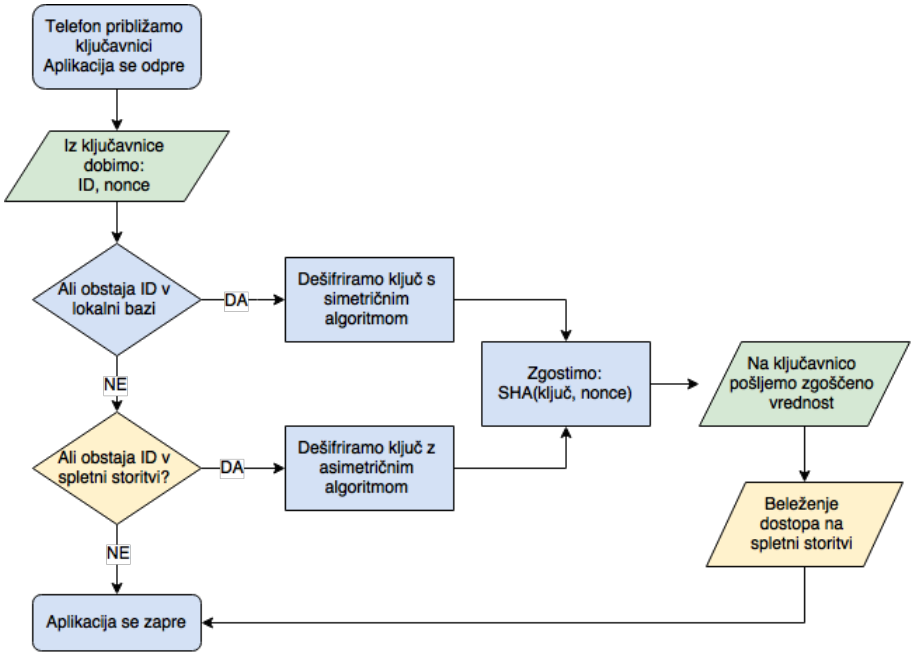
Preden začnemo uporabljati ključavnico, moramo v aplikaciji prijaviti ključavnico in vnesti podatke iz NFC značke. To storimo tako, da značko približamo telefonu, ta jo prepozna in shrani ključavnico. Ključavnici dodelimo tudi opisno ime in ikono. Ko ključavnico dodamo, lahko NFC značko pospravimo. Spet jo potrebujemo le ob izgubi ključa ali dodelitvi novega stalnega dostopa (glej poglavje 4.3.5).

Podroben potek prijave ključavnice je opisan v poglavju 4.3.3.

3.5 Odklep

Ko mobilni telefon približamo elektronski ključavnici, jo telefon zazna in zažene aplikacijo za odklepanje. Ta prebere NDEF sporočilo, ki vsebuje naključno generirano število in identifikacijsko številko ključavnice. Če v lokalni podatkovni bazi obstaja ključavnica s podanim identifikacijskim številom, prebrano naključno število zgosti z glavnim ključem te ključavnice in ga pošlje nazaj. Če ključavnica na obstaja v lokalni podatkovni bazi, preverimo še spletno storitev za primer da imamo dodeljen začasni dostop.

Ključavnica nato primerja prejeto zgoščeno vsebino z zgoščeno vsebino v pomnilniku in ob ujemanju izvede odklep. Po odklepu telefon spletni stori-



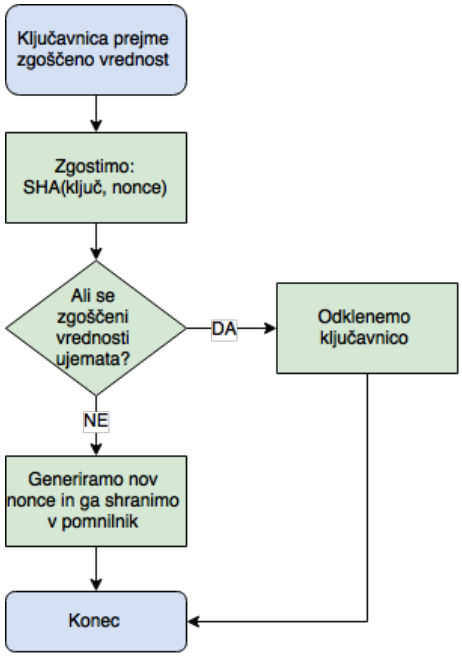
Slika 3.2: Diagram poteka za odklep ključavnice iz perspektive mobilne aplikacije

tvi sporoči uporabniško ime, id ključavnice, čas odklepa in GPS koordinate lokacije odklepa.

Na sliki 3.2 je prikazan diagram poteka odklepa iz perspektive mobilne aplikacije. Na sliki 3.3 pa je prikazan diagram poteka odklepa iz perspektive ključavnice.

3.6 Menjava ključa

Včasih se nam zgodi tudi, da ključe izgubimo ali pa nam jih ukradejo. Enako se lahko zgodi tudi z mobilnim telefonom, ki je v tem primeru hkrati ključ. V takšnem primeru lahko ključavnici zamenjamo glavno geslo. Ko v mobilni aplikaciji označimo, da želimo zamenjali geslo, se to avtomatsko generira. Telefon nato približamo znački in ključavnici ter zapišemo novo geslo, hkrati pa moramo poslati tudi staro geslo. Podrobnejši opis je v poglavju 4.3.5.



Slika 3.3: Diagram poteka za odklep ključavnice iz perspektive ključavnice

3.7 Začasni dostop

Ključavnico lahko uporabimo za različne namene in lahko se zgodi, da bi si prijatelj rad sposodil naše kolo, nas pa ni kje v bližini ali pa nas dlje časa ni doma in mora nekdo ta čas skrbeti za naš dom. Če imamo navadne ključavnice, moramo narediti kopijo ključa ali pa posoditi svojega.

Naša rešitev omogoča dodeljevanje začasnega dostopa drugim uporabnikom. Dodelimo lahko poljubno časovno obdobje, tako da izberemo začetni in končni datum. Ko uporabnik zahteva odklep ključavnice, aplikacija pri spletni storitvi preveri, ali ima ta pravico do začasnega dostopa. Primer začasnega dostopa je prikazan na sliki 3.2.

3.8 Zgodovina dostopov

Vsakič, ko ključavnico odklenemo se na spletno storitev shranijo podatki o trenutnem dostopu. Shranijo se podatki ključavnice, uporabnika, stanje baterije in GPS koordinate. Če GPS lokacija ni na voljo, se ta ne beleži. Zgodovine dostopov zaradi transparentnosti ni možno brisati. Dostop do podatkov o zgodovini ima samo lastnik ključavnice in ga ne omogoči gostom z začasnim dostopom.

Poglavje 4

Implementacija

4.1 Elektronska ključavnica

4.1.1 Končni avtomat

Koda za delovanje mikrokrmilnika na elektronski ključavnici je napisana v programskem jeziku C. Delovanje je določeno z determinističnim končnim avtomatom (ang. Finite State Machine) z naborom petih stanj. Prehodi med njimi so opisani na sliki 4.1. Opis posameznih stanj:

Čakanje - čakanje na prekinitev (ang. interrupt):

Mikrokrmilnik čaka na prekinitev iz zunanjega vira. Prekinitev se sproži ob koncu zapisa NDEF sporočila na NFC značko RF430CL330H. Ta prekinitev je nastavljena v registru za prekinitve, ki se nahaja na naslovu `0xFFFA`, na drugem najmanj pomembnem bitu (ang. least significant bit). Iz tega stanja je, odvisno od prejetega sporočila, možen prehod v stanje **Primerjava** oziroma **Menjava ključa**.

Primerjava - primerjava prejetega ključa s shranjenim:

Mikrokrmilnik iz NFC značke preko SPI vodila prebere vsebino pomnilnika (NDEF sporočilo), iz katerega nato izlušči zgoščen ključ. Iz svojega pomnilnika nato prebere pravilen ključ, ki ga zgosti in primerja

s prejetim. Če se ključa ujemata, se končni avtomat premakne v stanje **Odklepanje**, drugače se premakne v stanje **Generiranje**.

Generiranje - generiranje naključnega števila:
V tem stanju mikrokrmilnik naključno generira število in ga shrani v svoj pomnilnik Flash. Naslednje stanje je vedno **Čakanje**.

Odklepanje - odklepanje ključavnice:
Po uspešni primerjavi ključev, se končni avtomat prestavi v stanje odklepanja. V tem stanju mikrokrmilnik poskrbi za odklepanje ključavnice. Naslednje stanje je vedno **Generiranje**.

Menjava ključa - menjava glavnega ključa za odklep:
Mikrokrmilnik v tem stanju zamenja ključ shranjen v pomnilniku z novim. Nov ključ prebere iz prejetega NDEF sporočila. Naslednje stanje je vedno **Generiranje**.

Vrednosti v pomnilniku

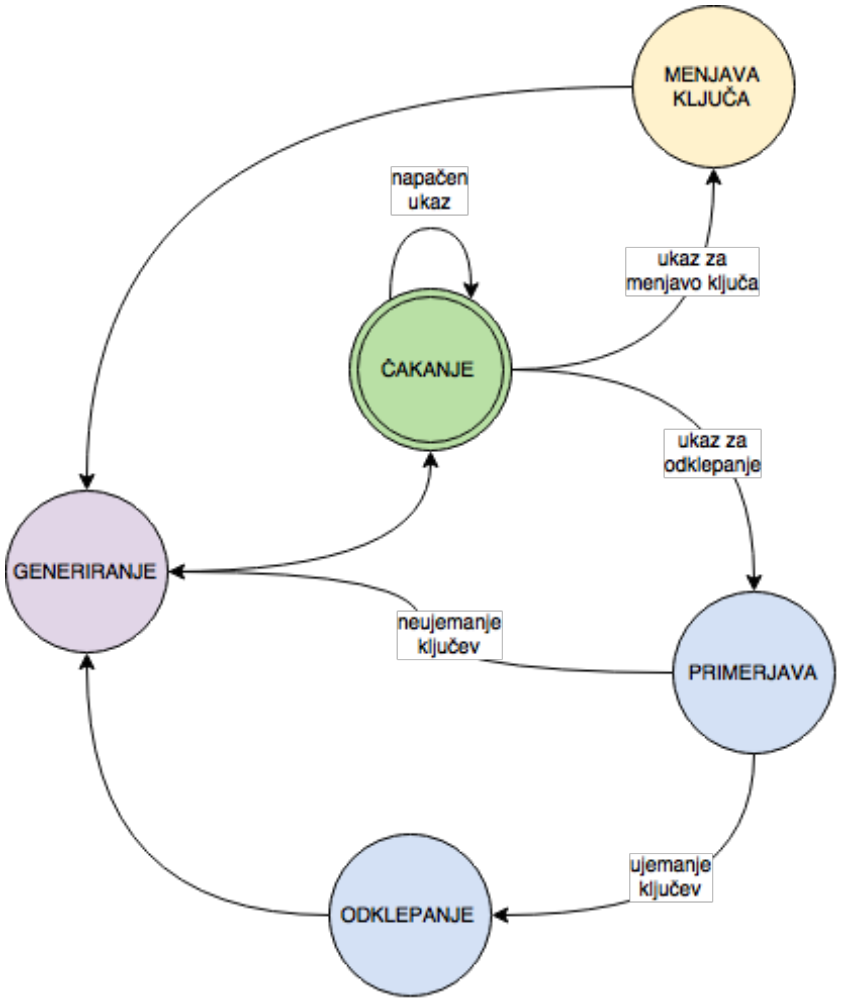
Vrednosti, ki so potrebne za delovanje in odklepanje ključavnice, so shranjene v Flash pomnilniku mikrokrmilnika. Lokacije in dolžine vrednosti v pomnilniku so opisane v tabeli 4.1.

Naslov	**Ime**	**Dolžina**
0x00	ID ključavnice	16 bajtov (128 bitov)
0x10	Ključ za odklepanje	16 bajtov (128 bitov)
0x20	Naključno generirano število	8 bajtov (64 bitov)

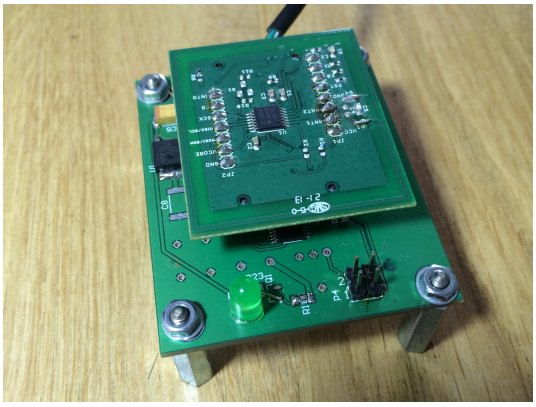
Tabela 4.1: Lokacije vrednosti v pomnilniku

4.1.2 Prototip

Strojni del prototipa elektronske ključavnice je rezultat diplomskega dela Jerneja Izaka z Univerze v Mariboru, z naslovom *Brezžična komunikacija*



Slika 4.1: Prehod med stanji končega avtomata



Slika 4.2: Prototip elektronske ključavnice

kratkega dosega za povezavo z mobilnimi napravami. Prototip je sestavljen iz dveh delov (slika 4.2): kontrolne plošče z mikrokrmilnikom ter dinamične NFC značke. Kontrolna plošča nam omogoča prikaz trenutnega stanja s pomočjo LED diode ali vodila USB (Universal Serial Bus).

4.2 Spletna storitev

Spletna storitev (ang. web service) je napisana v programskem jeziku Java in se izvaja na aplikacijskem strežniku GlassFish. Podatkovno podporo spletni storitvi zagotavlja podatkovna baza Java DB, ki je del paketa Glassfish.

Aplikacijski strežnik Glassfish je nameščen na virtualiziranem Linux strežniku in je dosegljiv na naslovu: `http://api.sharelock.ml:8080`.

4.2.1 Podatkovna baza

Podatkovna baza za podporo spletni storitvi je sestavljena iz šestih entitet in sedmih relacij. Entitetno-Relacijski (ER) model je predstavljen na sliki 4.3. Opis entitet:

Users predstavlja registrirane uporabnike. Vsebuje podatke o imenu, elektronskem naslovu in varnostne podatke. Vsebuje tudi naključno gene-

riran žeton.

Locks predstavlja vse ključavnice. Vsebuje ID, ime in številko ikone za prikaz v aplikaciji.

RegisteredLocks predstavlja registrirane ključavnice, ki jih uporabnik doda z aplikacijo. Poleg IDjev uporabnika in ključavnice se shrani še šifriran ključ za odklep ključavnice. Ta entiteta nam zagotavlja, da lahko eno ključavnico uporablja več uporabnikov.

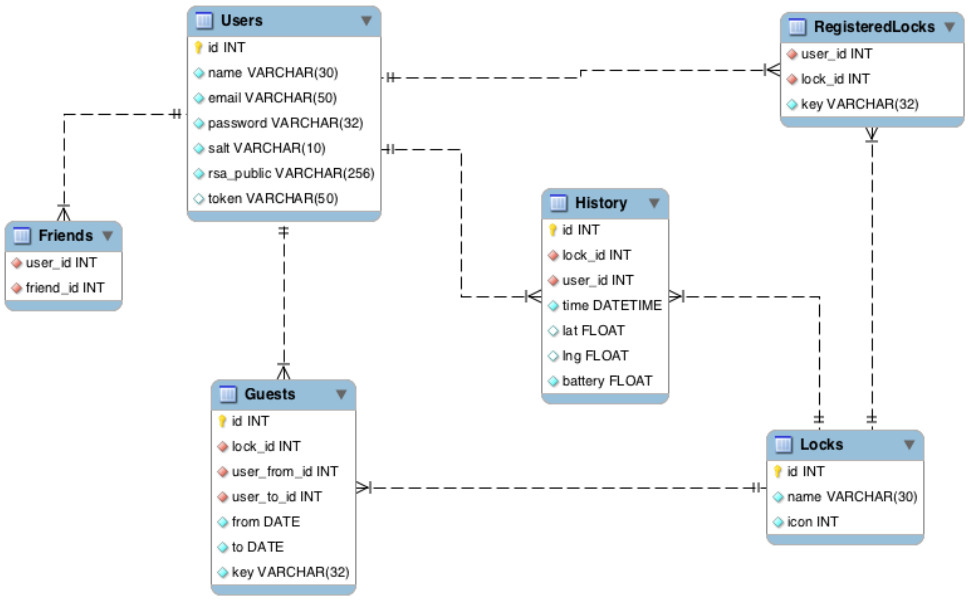
History predstavlja zgodovino ključavnice. Vsebuje ID uporabnika, ki je odklenil ključavnico, ID ključavnice, datum in čas dostopa, stanje o bateriji (v rangu od 0 do 1) ter neobvezna polja za zemljepisno širino in dolžino.

Guests predstavlja dodeljene začasne dostope za podano ključavnico. Poleg IDja ključavnice vsebuje še ID uporabnika, ki je dodelil dostop in ID uporabnika, ki mu je dodeljen. Določena sta še začetni in končni datum dodelitve in asimetrično šifriran ključ za odklep.

Friends predstavlja uporabnikove prijatelje, kjer je polje `user_id` ID uporabnika, `friend_id` pa ID prijatelja. Da lahko uporabniku dodelimo začasni dostop mora biti na našem seznamu prijateljev.

4.2.2 Avtentikacija z žetonom

Spletna storitev uporabnika uporablja princip avtentikacije uporabnika z žetonom. Žeton se generira ob prijavi uporabnika v sistem in se shrani v podatkovno bazo poleg uporabnika ter vrne kot odgovor. Ta žeton spletna storitev prejme ob vsakem nadaljnem zahtevku in ga preveri z žetonom v bazi. Če ima uporabnik, ki je povezan s tem žetonom ustrezne pravice, se zahtevek izvrši. V nasprotnem primeru spletna storitev vrne odgovor s kodo 401 - nepooblaščen dostop (ang. unauthorized). Primer takšne komunikacije med mobilno aplikacijo in strežnikom je prikazan na sliki 4.4.



Slika 4.3: ER diagram podatkovne baze

4.2.3 Java Beans objekti

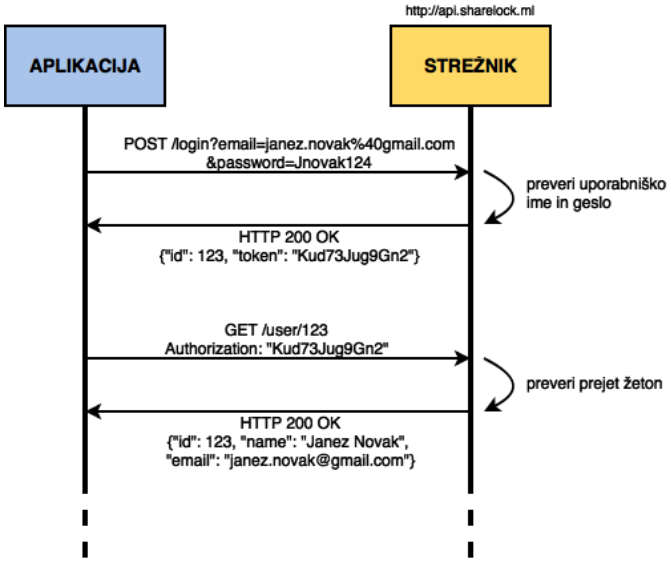
V spletni storitvi so uporabljeni štirje Java Beans objekti s podanimi lastnostmi (slika 4.5). Vsaki podani lastnosti pripadata še metodi za pridobivanje in nastavljanje (ang. getter, setter methods).

4.2.4 Storitve

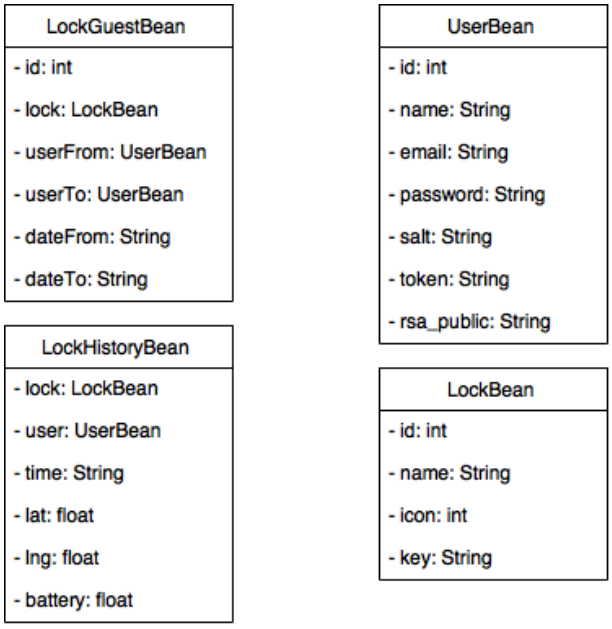
Storitve zagotavljajo štirje Javanski razredi, kjer je vsak dostopen na svojem url podnaslovu.

`UserService (/users)` - upravljanje z uporabniki:

- `/users`
  pridobivanje seznama uporabnikov
- `/users/register`
  registracija novega uporabnika



Slika 4.4: Primer komunikacije med mobilno aplikacijo in strežnikom



Slika 4.5: Uporabljeni Java Beans objekti

- `/users/login`
  prijava uporabnika v sistem
- `/users/{id}`
  pridobivanje podatkov uporabnika, posodabljanje podatkov
- `/users/{id}/avatar`
  nastavljanje prikazne slike
- `/users/{id}/locks`
  pridobivanje seznama ključavnic, dodajanje ključavnice
- `/users/{id}/friends`
  pridobivanje seznama prijatevljev
- `/users/{id}/friends/{friend_id}`
  pridobivanje določenega prijatelja, brisanje in dodajanje prijatelja

`HistoryService (/history)` - upravljanje z zgodovino ključavnic:

- `/history/{lock_id}`
  beleženje dostopov, pridobivanje dostopov
- `/history/{lock_id}/last`
  pridobivanje zadnjega dostopa

`LockService (/locks)` - upravljanje s ključavnicami:

- `/locks/`
  dodajanje ključavnice
- `/locks/{id}`
  urejanje in brisanje ključavnice

`GuestService (/guest)` - upravljanje z začasnimi dostopi:

- `/guests`
  dodajanje začasnega dostopa
- `/guests/{id}`
  brisanje začasnega dostopa

- `/guests/lock/{id}`
  pridobivanje začasnih dostopov za ključavnico
- `/guests/user/{id}`
  pridobivanje začasnih dostopov za uporabnika
- `/guests/lock/{lock_id}/user/{user_id}`
  pridobivanje ključa za začasni dostop

4.3 Mobilna aplikacija

Mobilna aplikacija deluje na pametnih telefonih z operacijskim sistemom Android verzije 4.1 ali novejših. Pogoj za delovanje aplikacije je podpora za NFC.

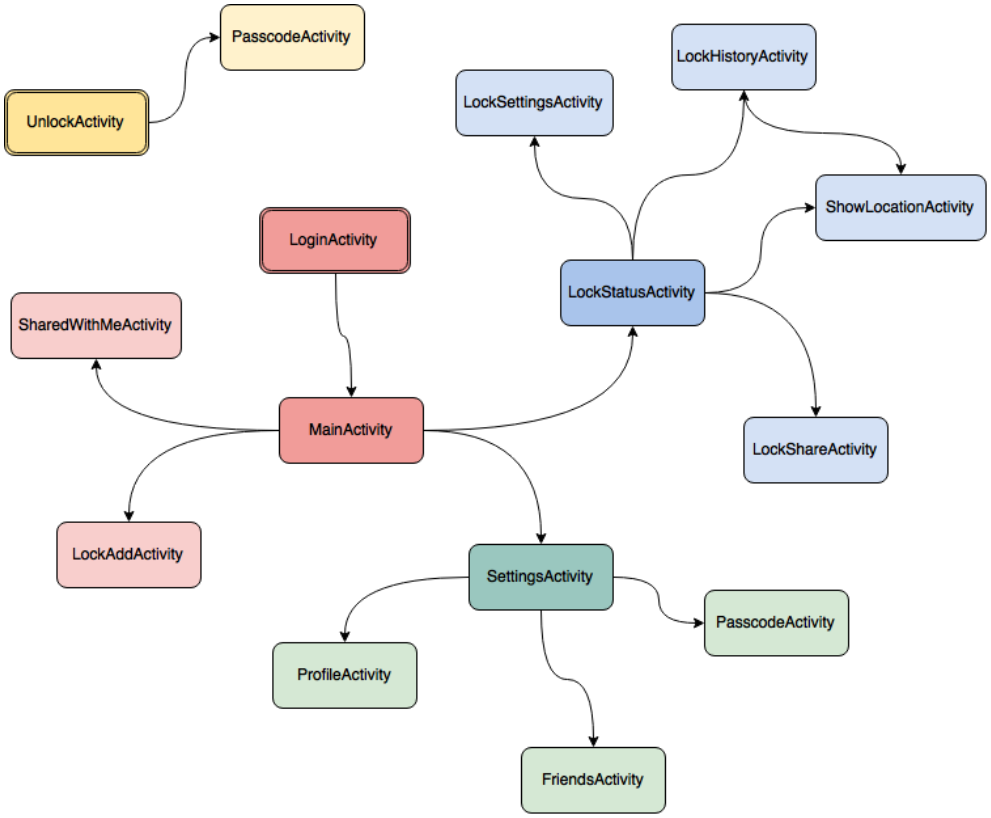
Mobilna aplikacije je sestavljena iz 14 aktivnosti, ki se med seboj povezujejo v celoto (slika 4.6). Logično jih lahko razdelimo na štiri dele:

1. Prijava/registracija uporabnika, glavni meni, dodajanje ključavnice (rdeča).
2. Nastavitve profila, prijateljev, PIN kode (zelena).
3. Prikaz stanja ključavnice, nastavitve, zgodovina, deljenje (modra).
4. Odklepanje ključavnice (rumena).

Aplikacija za samo odklepanje ključavnice ne potrebuje internetne povezave, medtem ko je za naprednejše funkcije le-ta potrebna. Med naprednejše funkcije spadajo ogled statusa ključavnice (dostopi, baterija, urejanje), omogočanje deljenja dostopa do ključavnice za določen čas in urejanje profila.

4.3.1 Prijava, registracija

Za prijavo in registracijo uporabikov je zadolžena aktivnost `LoginActivity` (slika 4.7). Pred prikazom obrazca za prijavo, aktivnost preveri, če je uporabnik že vpisan v sistem. Če je že vpisan, se pri spletni storitvi preverijo

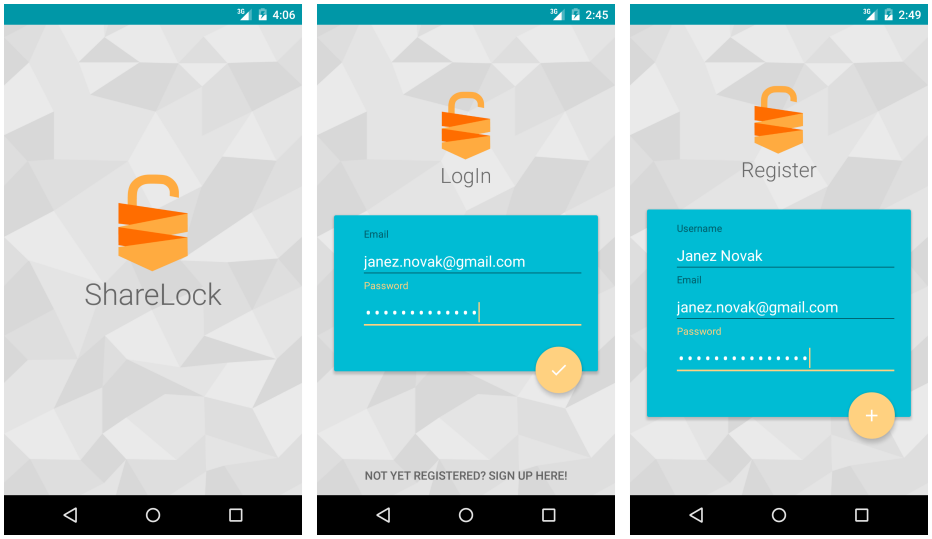


Slika 4.6: Potek aktivnosti. Logično povezane aktivnosti so združene z barvami.

uporabnikovi podatki. Če je prijava uspešna, spletna storitev vrne generiran žeton, ki ga shranimo v spomin, ker ga potrebujemo ob vsaki nadaljni poizvedbi s spletno storitvijo zaradi avtentikacije uporabnika. V primeru, da še ni vpisan (ali pa je bil izpisan) se prikaže obrazec za prijavo.

Obrazec za prijavo vsebuje polji za email in geslo uporabnika. Ob kliku na gumb za prijavo se najprej preveri njuna veljavnost. Email se preveri z regularnim izrazom, ki mora biti pravilne oblike, medtem ko mora biti geslo dolgo najmanj 6 znakov. V primeru napačnega vnosa se pod vnosom izpiše napaka. Če sta oba vnosa pravilna, se ponovno preverijo podatki pri spletni storitvi.

Pod obrazcem za prijavo je tudi gumb, ki odpre obrazec za registracijo

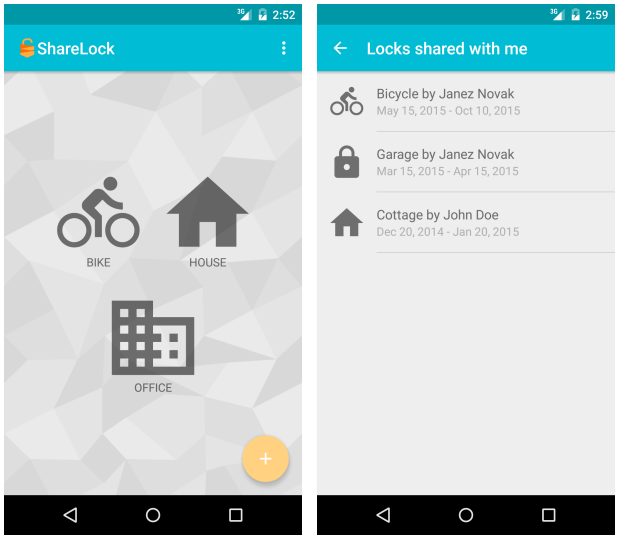


Slika 4.7: Prijava in registracija (`LoginActivity`)

novega uporabnika (slika 4.7 v sredini). Ta obrazec vsebuje poleg polj za email in geslo tudi polje za ime novega uporabnika. Enako kot pri prijavi, se tudi tu preverja veljavnost nizov. Ob kliku na gumb za registracijo pošljemo spletni storitvi podatke o novem uporabniku. V primeru pozitivnega odgovora (koda 202) lahko uspešno prijavimo uporabnika. V nasprotnem primeru (koda 200) izpišemo, da uporabnik že obstaja.

Avtentikacija z žetonom

Ko ob uspešni prijavi dobimo od spletne storitve generiran žeton (glej poglavje 4.2.2), moramo posodobiti razred za komunikacijo s spletno storitvijo. Od prijave naprej se mora v glavi (ang. header) vsakega HTTP zahtevka prenesti tudi žeton. To storimo s klicem statične metode `createService(String)` razreda `ApiManager`, ki nastavi fiksno glavo vsem nadaljnim zahtevkom.



Slika 4.8: `MainActivity` in `SharedWithMeActivity`

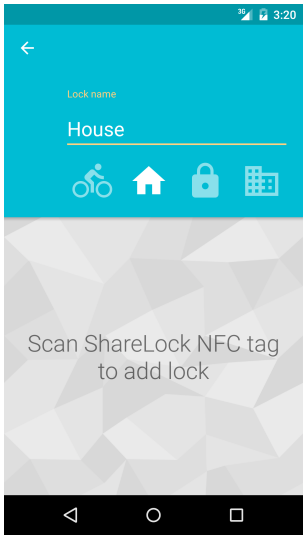
4.3.2 Glavni meni

Po uspešni prijavi se odpre glavni meni (slika 4.8 levo), katerega predstavlja aktivnost `MainActivity`. Pri odprtju aktivnosti se najprej iz spletne storitve preberejo ključavnice, ki jih že imamo dodane. Te ključavnice se nato izpišejo v ospredju s pripadajočo ikono in imenom. V primeru, da še nimamo dodanih ključavnic se izpiše navodilo.

S pritiskom na ključavnico, se odpre aktivnost s stanjem izbrane ključavnice. S pritiskom na gumb v desnem spodnjem kotu, pa se odpre aktivnost za dodajanje novih ključavnic. V meniju imamo tudi možnost odprtja aktivnosti za nastavitve in pregleda nam dodeljenih dostopov do ključavnic (slika 4.8 desno).

Sinhronizacija z lokalno bazo

Da lahko ključavnico odklenemo tudi brez aktivne internetne povezave, se morajo ključavnice prebrane iz spletnega servisa sinhronizirati z lokalno podatkovno bazo SQLite. Sinhronizirata se le polja ID in ključ, ki sta potrebna za odklep ključavnice. Ključ, ki se shrani lokalno, je še vedno šifriran in ga



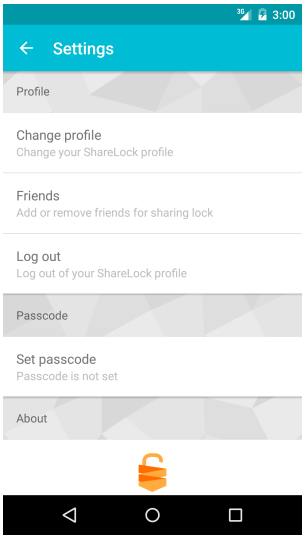
Slika 4.9: `LockAddActivity`

dešifriramo šele pri odklepu.

4.3.3 Dodajanje ključavnice

Za dodajanje ključavnic uporabnika je odgovorna aktivnost `LockAddActivity`. Za njeno dodajanje potrebujemo nekaj podatkov o njej. Ime in ikono ključavnice nastavi uporabnik sam (slika 4.9), medtem ko moramo pridobiti ID in ključ za odklep z branjem iz priložene NFC značke. Ko uporabnik skenira NFC značko, se iz značke preberejo podatki in v desnem spodnjem kotu se prikaže gumb za dodajanje "+". Ob kliku na ta gumb se najprej preveri veljavnost vnešenega imena, ki mora biti daljše od 5 znakov. Če je ime veljavno, se na spletno storitev pošlje zahtevek za dodajanje ključavnice. V primeru uspešno dodane ključavnice (koda 201) se aktivnost zapre, drugače se izpiše ustrezna napaka.

Aktivnost omogoča branje le pravilno formatiranih NFC značk. Značka mora vsebovati NDEF sporočilo tipa `External`, predpono `com.mj.sharelock.lock` in dva zapisa: ID in ključ.



Slika 4.10: `SettingsActivity`

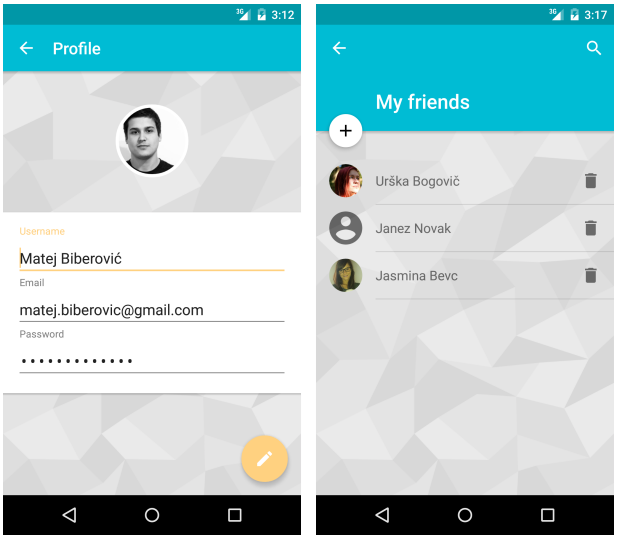
4.3.4 Nastavitve

V aktivnosti za nastavitve `SettingsActivity` (slika 4.10) imamo dostop do nastavitev profila trenutnega uporabnika. Ustvarimo lahko tudi štirimestno PIN kodo, za dodatno varnost. Na dnu pa so prikazani podatki o programu in avtorju.

Profil

S klikom na gumb "Change profile" se odpre aktivnost `ProfileActivity` (slika 4.11 levo), kjer lahko posodobimo podatke o trenutnem uporabniškem profilu. S klikom na sliko (ang. avatar), lahko zamenjamo trenutno sliko uporabnika. Odpre se fotoaparat, s katerim lahko zajamemo novo sliko. S klikom na gumb za urejanje se preveri veljavnost vnešenih podatkov. Ob pravilnem vnosu se na spletno storitev pošlje posodobljen profil.

S klikom na gumb "Friends" se odpre aktivnost `FriendsActivity` (slika 4.11 desno), kjer imamo vpogled v seznam uporabnikov aplikacije, ki so naši prijatelji. Da lahko uporabniku dodamo začasni dostop do ključavnice, mora biti naš prijatelj. Nove prijatelje dodajamo s klikom na gumb "+", ki nam



Slika 4.11: `ProfileActivity` in `FriendsActivity`

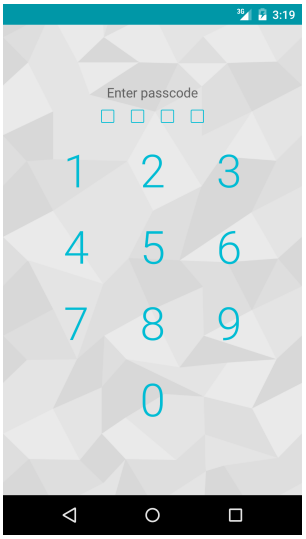
odpre iskalnik po vseh uporabnikih. Iščemo lahko po uporabniških imenih ali email naslovih.

PIN koda za odklep

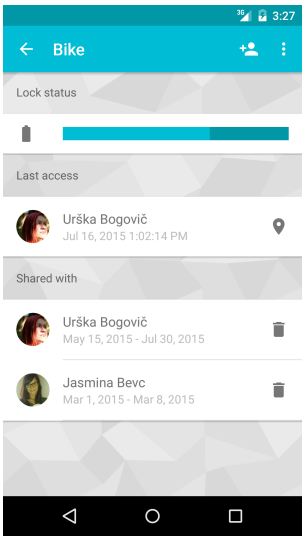
S klikom na gumb "Set passcode" oziroma "Clear passcode" se odpre aktivnost `PasscodeActivity`. Tu lahko za dodatno varnost pri odklepu ključavnice nastavimo PIN kodo (slika 4.12). Za nastavitev moramo dvakrat zaporedi vnesti poljubno štirimestno kodo. Če se ti dve zaporedji ujemata, se PIN koda shrani za nadaljno primerjavo. Ko želimo naslednjič odkleniti ključavnico, moramo pred odklepom vnesti pravilno PIN kodo. V primeru, da je PIN koda že nastavljena, se s klikom na ta gumb preverjanje le-te izklopi.

4.3.5 Upravljanje ključavnice

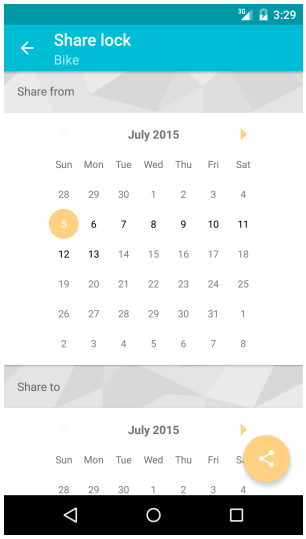
Upravljanje ključavnice nam omogoča vpogled v stanje izbrane ključavnice in nastavljanje dodatnih možnosti.



Slika 4.12: `PasscodeActivity`



Slika 4.13: `LockStatusActivity`



Slika 4.14: `LockShareActivity`

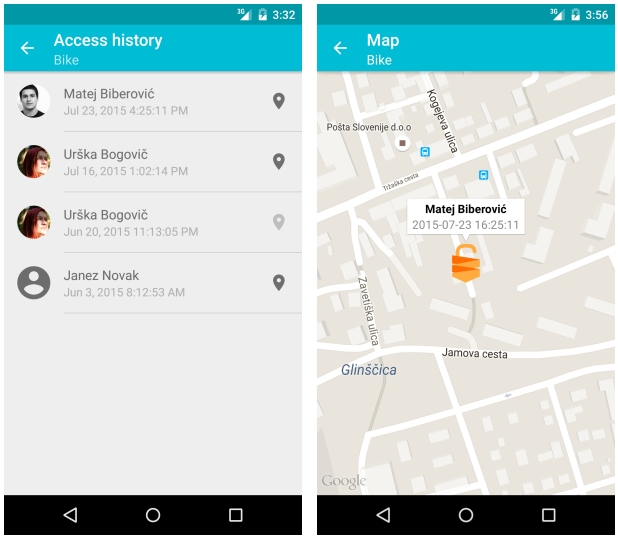
Prva aktivnost `LockStatusActivity` predstavlja trenutno stanje ključavnice (slika 4.13). Na voljo so podatki:

- baterija: delež preostanka kapacitete,
- zadnji dostop: uporabnik, čas, lokacija,
- začasni dostop: uporabnik, čas od-do, možnost izbrisa.

Tu imamo tudi možnost deljenja dostopa do ključavnice drugim uporabnikom. V meniju pa lahko odpremo izpis celotne zgodovine ali nastavitev trenutne ključavnice.

Deljenje dostopa

Ena izmed glavnih funkcij aplikacije je deljenje dostopa do ključavnice drugim uporabnikom. To nam omogoča aktivnost `LockShareActivity` (slika 4.14). Tu z izbiro začetnega in končnega datuma določimo obdobje dodeljenega dostopa. Ko izberemo končni datum, se nam prikaže gumb za deljenje. Ob kliku na ta gumb lahko izbiramo med dodanimi prijatelji.



Slika 4.15: `LockHistoryActivity` in `LockMapActivity`

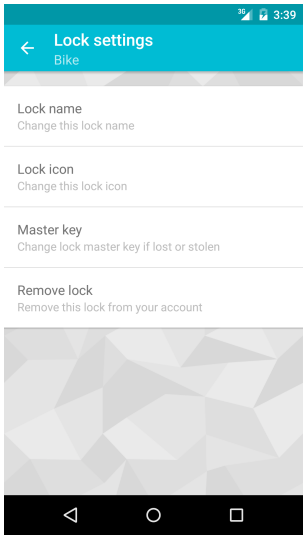
Ogled zgodovine

Aktivnost s stanjem ključanice nam prikaže le zadnji dostop, medtem ko aktivnost `LockHistoryActivity` prikaže seznam vseh dostopov do ključavnice (slika 4.15 levo). Teh dostopov ni možno izbrisati, če pa imajo dodeljeno lokacijo (zemljepisno širino in dolžino) pa si lahko ogledamo podrobno lokacijo na zemljevidu (slika 4.15 desno). Na zemljevidu se prikaže označena lokacija skupaj z imenom uporabnika in časom dostopa. Za prikaz zemljevida so uporabljene karte Google Maps.

Nastavitve ključavnice

Vsaka ključavnica ima tudi možnost spremembe nastavitev v aktivnosti `LockSettingsActivity` (slika 4.16). Ključavnici lahko spremenimo ime, ikono ali v primeru izgube telefona spremenimo glavni ključ. Pri kliku na spremembo imena ali ikone se odpre pogovorno okno (ang. dialog window), kjer lahko posodobimo podatke. Ključavnico lahko tudi odstranimo, če je ne potrebujemo več.

S klikom na gumb "Master key"lahko spremenimo glavni ključ za odkle-



Slika 4.16: `LockSettingsActivity`

panje ključavnice. S pomočjo javanskega razreda `UUID` generiramo nov ID in ključ. Ta par kot NDEF sporočilo zapišemo na NFC značko in na elektronsko ključavnico. Po uspešnem zapisu, se posodobi tudi vrednost na spletni storitvi.

4.3.6 Odklepanje

Za glavni del aplikacije, ki omogoča odklep ključavnice skrbi aktivnost `UnlockActivity`. Ta aktivnost nima uporabniškega vmesnika in se zažene takrat ko telefon približamo ključavnici. Ključavnica vsebuje NDEF sporočilo s tremi podatki: ID ključavnice, stanje o bateriji in enkratno generirano število.

Aktivnost najprej preveri, če imamo nastavljeno PIN kodo. Če je leta nastavljena, jo moramo pred nadaljevanjem vnesti. V naslednji fazi se preveri, če je ključavnica s podanim IDjem na voljo v lokalni podatkovni bazi SQLite. Če obstaja v bazi, lahko nadaljujemo. V primeru, da ključavnice ni v lokalni bazi, preverimo še v spletni storitvi, če imamo mogoče dodeljen začasni dostop. Če dobimo od spletne storitce pozitiven odgovor (koda 200)

in ključ, lahko nadaljujemo.

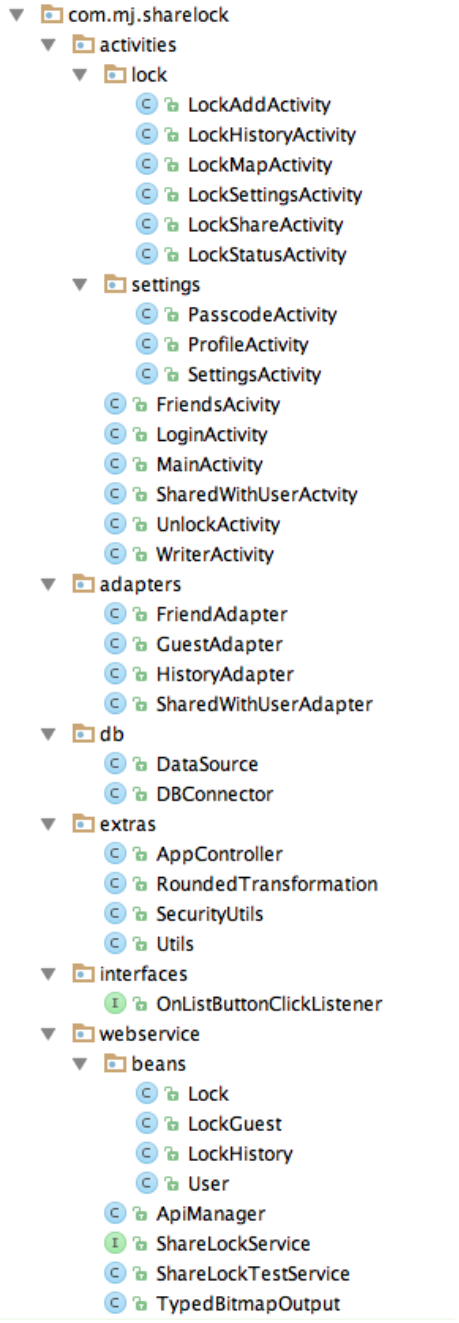
Če pridobimo ključ za odklepanje iz lokalne baze SQLite, ga moramo najprej dešifrirati z AES algoritmom, kjer je uporabnikovo geslo ključ za dešifriranje. Za dešifriranje se uporablja razred `Cipher`. Če pa pridobimo ključ iz spletne storitve, ključ dešifriramo z zasebnim RSA ključem, shranjenim v pomnilniku.

Dešifriran ključ in prejeto enkratno generirano število zgostimo z algoritmom SHA-256 in zapišemo v NDEF sporočilo. NDEF sporočilo se nato prenese na ključavnico.

4.3.7 Datoteke

Za delovanje mobilne aplikacije skrbi 32 Javanskih razredov in 2 Javanska vmesnika (slika 4.17). Opis posameznih datotek in map:

- **activities**: aktivnosti z uporabniškim vmesnikom so opisane v prejšnjih poglavjih.
- **adapters**: pomožni razredi za polnjenje seznamov (prijetelji, dodeljeni dostopi, zgodovina).
- **db**: pomožni razredi za kreiranje in dostop do lokalne SQLite podatkovne baze.
- **extras**: ostali pomožni razredi: dostop do shranjenih nastavitev, metode za pretvarjanje in šifriranje, ipd.
- **interfaces**: vmesnik za lovljenje klikov na elemente seznama.
- **webservice**: razredi za pomoč pri povezovanju s spletno storitvijo.
- **beans**: razredi za prenos podatkov med telefonom in spletno storitvijo.



Slika 4.17: Uporabljeni Javanski razredi in vmesniki

Poglavje 5

Sklepne ugotovitve

V diplomskem delu smo predstavili sistem, ki nam omogoča odklepanje elektronske ključavnice s pomočjo pametnega mobilnega telefona in tehnologije NFC. Opisali smo tehnologije in orodja, definirali delovanje in predstavili konkretno implementacijo sistema. Za prikaz delovanja smo razvili tudi enostaven prototip, ki bi bil z nekaj popravki in izboljšavami že pripravljen za komercialno uporabo. V diplomskem delu smo tako pokazali praktično uporabo tehnologije NFC in izzive, s katerimi smo se med tem soočili.

Največji izziv pri razvoju je bila uspešna implementacija varnostnih mehanizmov, ki so pri sistemih za dostop ključnega pomena. Za varnost smo poskrbeli na večih področjih, vendar so pomankljivosti še vedno prisotne:

Uporabniško geslo in PIN koda za odklep sta shranjena v pomnilniku telefona. Načeloma ima dostop do teh podatkov samo aplikacija, s katero smo te podatke shranili. Edini problem se pojavi, če je uporabnik eksplicitno pridobil administratorske pravice (ang. root access) in nam takšen telefon odtujijo. Morebitnemu napadalcu bi lahko delo otežili s šifriranjem uporabniškega imena in gesla, vendar bi iz izvorne kode aplikacije lahko še vedno pridobil ključe za dešifriranje.

Glavni ključ za odklep ključavnice je shranjen na večih mestih. V pomnilniku NFC značke za registracijo nove ključavnice je ključ shranjen

kot čistopis, zato moramo to NFC značko skrbno varovati. V pomnilniku telefona in v podatkovni bazi pa je ključ šifriran z uporabniškim geslom. Da preprečimo zlorabo glavnega ključa, mora zato uporabnik izbrati dovolj zapleteno geslo. Morebitni napadalec bi lahko pridobil tudi fizični dostop do mikrokrmilnika na elektronski ključavnici in prebral shranjen ključ ali pa celo spremenil programsko kodo.

Omogočanje začasnega dostopa nam je predstavljalo nov problem, ker je potrebno prenesti glavni ključ do drugega uporabnika. Ta ključ je shranjen v podatkovni bazi spletne storitve in je šifriran z javnim ključem drugega uporabnika. Da ta uporabnik ne bi videl glavnega ključa v obliki čistopisa, se zgoščevanje ključa z naključnim številom izvede v spletni storitvi. Če bi napadalec prevzel nadzor nad našim spletnim strežnikom, bi lahko prestregel ta ključ.

Komunikacija med telefonom in spletno storitvijo trenutno poteka s pomočjo protokola HTTP v čistopisu. Napadalec bi lahko prestregel promet in iz njega izluščil uporabniška gesla in glavne ključe. Da bi se temu izognili, bi morali pri overitelju digitalni potrdil (npr. DigiCert) kupiti digitalno potrdilo in za komunikacijo uporabiti HTTPS protokol.

Zamenjava glavnega ključa nam predstavlja problem, če se posodobljeni podatki niso uspešno prenesli na elektronsko ključavnico. Tako lahko pride do okvare podatkov in izgubimo dostop do ključavnice. To bi lahko rešili s ponovnim preverjanjem zapisanih podatkov, na račun zahtevnejšega postopka zamenjave.

Sistem smo testirali tudi z različnimi vrstami naprav z operacijskim sistem Android. Aplikacija je na vseh testiranih telefonih delovala dobro. Razlike so bile le v dometu, ki pa je odvisen od dimenzij antene in vrste NFC vgrajenega sistema v telefonu. Pomanjkljivost je tudi omejitev delovanja aplikacije na operacijskem sistemu Android. Operacijski sistem iOS sicer ima podporo za NFC, vendar zaenkrat ta še ni na voljo razvijalcem in je uporabljen le za brezgotovinsko plačevanje ApplePay.

Literatura

[1] K. Curran, A. Millar, C. McGarvey. *Near Field Communication* [Online]. Dosegljivo: http://www.iaesjournal.com/online/index.php/IJECE/article/view/234

[2] NFC Forum - Tag Type Technical Specifications [Online]. Dosegljivo: http://nfc-forum.org/our-work/specifications-and-application-documents/specifications/tag-type-technical-specifications/

[3] Replay Attacks by Ted Hersey [Online]. Dosegljivo: http://all.net/CID/Attack/papers/Replay.html.

[4] E. Haselsteiner, K. Breitfuß. *Security in Near Field Communication (NFC): Strengths and Weaknesses*. [Online]. Dosegljivo: http://events.iaik.tugraz.at/RFIDSec06/Program/papers/002%20-%20Security%20in%20NFC.pdf

[5] NFC Forum [Online]. Dosegljivo: http://nfc-forum.org

[6] NFC Forum - NFC Data Exchange Format (NDEF) - Technical Specification [Online]. Dosegljivo: http://www.eet-china.com/ARTICLES/2006AUG/PDF/NFCForum-TS-NDEF.pdf

[7] REST API Tutorial - What Is REST? [Online]. Dosegljivo: http://www.restapitutorial.com/lessons/whatisrest.html

[8] M. Owens. *The Definitive Guide to SQLite* [Online]. Dosegljivo: http://read.pudn.com/downloads109/doc/451443/The%20Definitive%20Guide%20To%20SQLite%20(2006).pdf

[9] Texas Instuments - MSP430G2231 datasheet [Online]. Dosegljivo: http://www.ti.com/lit/ds/symlink/msp430g2231.pdf

[10] Texas Instruments - RF430CL330H Dynamic NFC Interface Transponder datasheet [Online]. Dosegljivo: http://www.ti.com/lit/ds/slas916c/slas916c.pdf

[11] Smartphone OS Market Share [Online]. Dosegljivo: http://www.idc.com/prodserv/smartphone-os-market-share.jsp

[12] Number of Android applications [Online]. Dosegljivo: http://www.appbrain.com/stats/number-of-android-apps

[13] E. Schaefer, Santa Clara University. *An introduction to cryptography and cryptanalysis* [Online]. Dosegljivo: http://math.scu.edu/ eschaefe/book.pdf

[14] Wireless Smart Door Locks [Online]. Dosegljivo: http://postscapes.com/wireless-door-locks

[15] J. Black, M. Cohran, T. Highland- *A Study of the MD5 Attacks: Insights and Improvements* [Online]. Dosegljivo: https://www.iacr.org/archive/fse2006/40470265/40470265.pdf